## DEEP LEARNING BASED CHRONIC KIDNEY

## DISEASE DIAGNOSIS

B. JANARDHAN

K. MOUNIKA

R. ASHWITHA



### **Department of Electronics and Communication Engineering**

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, T.S.)

Chaitanya Bharathi P.O., Gandipet, Hyderabad – 500 075

2023

## DEEP LEARNING BASED CHRONIC KIDNEY

## DISEASE DIAGNOSIS

MAJOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

BY

B. JANARDHAN (19261A0406)

K. MOUNIKA (19261A0426)

R. ASHWITHA (19261A0451)



Department of Electronics and Communication Engineering

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

#### (Affiliated to Jawaharlal Nehru Technological University, Hyderabad, T.S.)

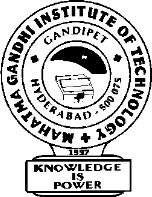
Chaitanya Bharathi P.O., Gandipet, Hyderabad – 500 075

2023

**MAHATMA GANDHI INSTITUTE OF TECHNOLOGY**

#### (Affiliated to Jawaharlal Nehru Technological University, Hyderabad, T.S)

Chaitanya Bharathi P.O., Gandipet, Hyderabad-500 075



**Department of Electronics and Communication Engineering**

**CERTIFICATE**

Date: 30 December 2022

This is to certify that the Mini project work entitled “**DEEP LEARNING BASED CHRONIC KIDNEY DISEASE DIAGNOSIS**” is a bonafide work carried out by

#### B. JANARDHAN (19261A0406) K. MOUNIKA (19261A0426)

**R. ASHWITHA (19261A0451)**

in partial fulfillment of the requirements for the degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS & COMMUNICATION ENGINEERING** by the

Jawaharlal Nehru Technological University, Hyderabad during the academic year 2022-23.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

(Signature) (Signature)

**Mr. D. V. S. Nagendra Kumar Dr. S P Singh**

**(Assistant professor)**

**Professor & Head of the department**

**Faculty Advisor/Liaison**

(i)

# **ACKNOWLEDMENT**

We wish to express our sincere thanks to Dr. S.P. Singh (Head of the Department) of Electronics and Communication Engineering and Dr.D.Venkat Reddy (Professor), M.G.I.T., for permitting us to pursue our Project in Cranes Varsity and encouraging us throughout the Project.

We wish to express our sincere thanks to Dr. S.P. Singh (Head of the Department) of Electronics and Communication Engineering and Dr.D.Venkat Reddy Professor, Dr.T.R.Vijaya Lakshmi Professor M.G.I.T., for permitting us to pursue our Project in Cranes Varsity and encouraging us throughout the Project.

Finally, we thank all the people who have directly or indirectly help us through the course of our Project.

B. JANARDHAN (19261A0406) K. MOUNIKA (19261A0426)

R. ASHWITHA (19261A0451)

(ii)

**ABSTRACT**

Chronic kidney disease (CKD) is a global health problem with high morbidity and mortality rate, and it induces other diseases. Since there are no obvious symptoms during the early stages of CKD, patients often fail to notice the disease. Early detection of CKD enables patients to receive timely treatment to ameliorate the progression of this disease. Machine learning models can effectively aid clinicians achieve this goal due to their fast and accurate recognition performance.

In this study, we propose a machine learning methodology for diagnosing CKD. The CKD data set was obtained from the University of California Irvine (UCI) machine learning repository, which has a large number of missing values. KNN imputation was used to fill in the missing values, which selects several complete samples with the most similar measurements to process the missing data for each incomplete sample. Missing values are usually seen in real-life medical situations because patients may miss some measurements for various reasons. We proposed an CNN and Mobilenet, best accuracy hence, we speculated that this methodology could be applicable to more complicated clinical data for disease diagnosis.

**Keywords**: - CNN, Mobilenet. CKD data set and Deep learning model

(iii)

# **Table of contents**

|  |  |
| --- | --- |
| **CERTIFICATE FROM ECE DEPARTMENT** | (i) |
| **ACKNOWLEDGEMENTS** | (ii) |
| **ABSTRACT** | (iii) |
| **LIST OF FIGURES** | (v) |
| **LIST OF TABLES** | (vi) |

**CHAPTER 1. OVERVIEW 1**

1.1 Introduction 1

1.2 Aim of the project 3

1.3 Methodology 4

1.3.1 Layers Used In Pre-Trained Model-Mobile Net 5

1.3.2 Pre-Trained Model-Mobile Net 12

1.4 Significance and applications 19

1.5 Organization of work 20

**CHAPTER 2. LITERATURE SURVEY 21**

2.1 Case studies 21

2.2 Existing System 23

2.3 Proposed System 23

**CHAPTER 3. SYSTEM REQUIREMENT SPECIFICATIONS 27**

3.1 Software Requirements And Hardware Requirements 27

**CHAPTER 4. SYSTEM ANALYSIS AND SYSTEM DESIGNS 31**

4.1 Feasibility Study 31

(iv)

4.2 System Design 32

**CHAPTER 5. RESULTS AND CONCLUSIONS 36**

5.1 Results 36

5.1.1 Results from CNN 36

5.1.2 Results from Pre-Trained Model 38

5.2 Conclusion and Future Scope 40

**LIST OF FIGURES**

1.1 BASIC PICTURE OF KIDNEYS…………………………………………………….1

1.2 PICTURE OF KIDNEY DIAGNOSIS………………………………………………..3

1.3 DIAGRAM OF FEATURE DETECTORFROM INPUT IMAGE…………………...5

1.4 FORMATION OF CONVOLUTION LAYER FROM INPUT IMAGE…...........…...6

1.5 IMAGE OF RELU LAYER FORMATION…………………………………………..6

1.6 MAX POOLING OPERATION……………………………………………………....8

1.7 AVERAGE POOLING OPERATION………………………………………………..8

1.8 FLATTENING LAYER FROM FEATURE MAP…………………………………...9

1.9 FLATTENED INPUT LAYER……………………………………………………….9

1.10 PICTURE OF FULLY CONNECTED LAYERS .…………………………………11

1.11 BLOCK DIAGRAM OF FULLY CONNECTED LAYERS……………………….12

1.12 CONNECTION OF LAYERS IN CNN…………………………………………….12

1.13 DIAGRAM OF PRE-TRAINED MODEL………………………………………….13

1.14 BLOCK DIAGRAM OF DEPTH WISE AND POINT WISE CONVOLUTION….14

1.15 BLOCK DIAGRAM OF DEPTH WISE CONVOLUTION………………………..15

1.16 BLOCK DIAGRAM OF POINT WISE CONVOLUTION………………………...16

1.17 BLOCK DIAGRAM OF CONVOLUTION UNIT…………………………………16

1.18 DIAGRAM OF SOFTWARE DEVELOPMENT LIFE CYCLE…………………..17

1.19 BLOCK DIAGRAM OF CNN ALGORITHM……………………………………..18

2.1 STEPS INVOLVED IN EXIXTING SYSTEM……………………………………..24

2.2 WORK FLOW OF THE PROJECT…………………………………………………25

3.1 PERFORMANCE COMPARISION ON TRAINING COST……………………….30

4.1 DIAGRAM OF DFD STEP 1………………………………………………………..33

4.2 DIAGRAM OF DFD STEP 2………………………………………………………..34

5.1 RESULT FROM CNN………………………………………………………………36

5.2 GRAPH OF RESULT FROM CNN………………………………………………...37

5.3 RESULT FROM PRE-TARINED MODEL………………………………………...38

5.4 GRAPH OF RESULT FROM PRE-TRAINED MODEL…………………………..39

(v)

**LIST OF TABLES**

4.1 TABLE OF TEST CASES……………………………………………………………34

4.2 TABLE OF TEST CASES MODEL BUILDING…………………………………….35

(vi)

**CHAPTER 1. OVERVIEW**

**1.1. INTRODUCTION**

CHRONIC kidney disease (CKD) is a global public health problem affecting approximately 10% of the world’s population. The percentage of prevalence of CKD in China is 10.8% and the range of prevalence is 10%-15% in the United States. According to another study, this percentage has reached 14.7% in the Mexican adult general population. This disease is characterized by a slow deterioration in renal function, which eventually causes a complete loss of renal function. CKD does not show obvious symptoms in its early stages. Therefore, the disease may not be detected until the kidney loses about 25% of its function. In addition, CKD has high morbidity and mortality, with a global impact on the human body. It can induce the occurrence of cardiovascular disease . CKD is a progressive and irreversible pathologic syndrome. Hence, the prediction and diagnosis of CKD in its early stages is quite essential, it may be able to enable patients to receive timely treatment to ameliorate the progression of the disease. Machine learning refers to a computer program, which calculates and deduces the information related to the task and obtains the characteristics of the corresponding pattern. This technology can achieve accurate and economical diagnoses of diseases; hence, it might be a promising method for diagnosing CKD. It has become a new kind of medical tool with the development of information technology and has a broad application prospect because of the rapid development of electronic health record. In the medical field, machine learning has already been used to detect human body status, analyze the relevant factors of the disease and diagnose various diseases. For example, the models built by machine learning algorithms were used to diagnose heart disease diabetes and retinopathy, acute kidney injury cancer and other diseases. In these models, algorithms based on regression, tree, probability, decision surface and neural network were often effective. In the field of CKD diagnosis, Hodnel and et al. utilized image registration to detect renal morphologic changes. Vasquez-Morales et al. established a classifier based on neural network using large-scale CKD data, and the accuracy of the model on their test data was 95% .In addition, most of the previous studies utilized the CKD data set that was obtained from the UCI machine learning repository. Chen et al. used k-nearest neighbor (KNN), support vector machine (SVM) and soft independent modelling of class analogy to diagnose CKD, KNN and SVM achieved the highest accuracy of 99.7%. In addition, they used fuzzy rule-building expert system, fuzzy optimal associative memory and partial least squares discriminant analysis to diagnose CKD, and the range of accuracy in those models was 95.5%-99.6%.

1

Their studies have achieved good results in the diagnosis of CKD. In the above models, the mean imputation is used to fill in the missing values and it depends on the diagnostic categories of the samples.

As a result, their method could not be used when the diagnostic results of the samples are unknown. In reality, patients might miss some measurements for various reasons before diagnosing. In addition, for missing values in categorical variables, data obtained using mean imputation might have a large deviation from the actual values. For example, for variables with only two categories, we set the categories to 0 and 1, but the mean of the variables might be between 0 and 1. Polat et al. developed an SVM based on feature selection technology, the proposed models reduced the computational cost through feature selection, and the range of accuracy in those models was from 97.75%-98.5% J. Aljaaf et al. used novel multiple imputation to fill in the missing values, and then MLP neural network (MLP) achieved an accuracy of 98.1% .Subas et al. used MLP, SVM, KNN, C4.5 decision tree and random forest (RF) to diagnose CKD, and the RF achieved an accuracy of 100%. In the models established by Boukenze et al., MLP achieved the highest accuracy of 99.75%. The studies of focus mainly on the establishment of models and achieve an ideal result. However, a complete process of filling in the missing values is not described in detail, and no feature selection technology is used to select predictors as well. Almansour et al. used SVM and neural network to diagnose CKD, and the accuracy of the models was 97.75% and 99.75%, respectively. In the models established by Gunarathne et al., zero was used to fill out the missing values and decision forest achieved the best performance with the accuracy was 99.1%.

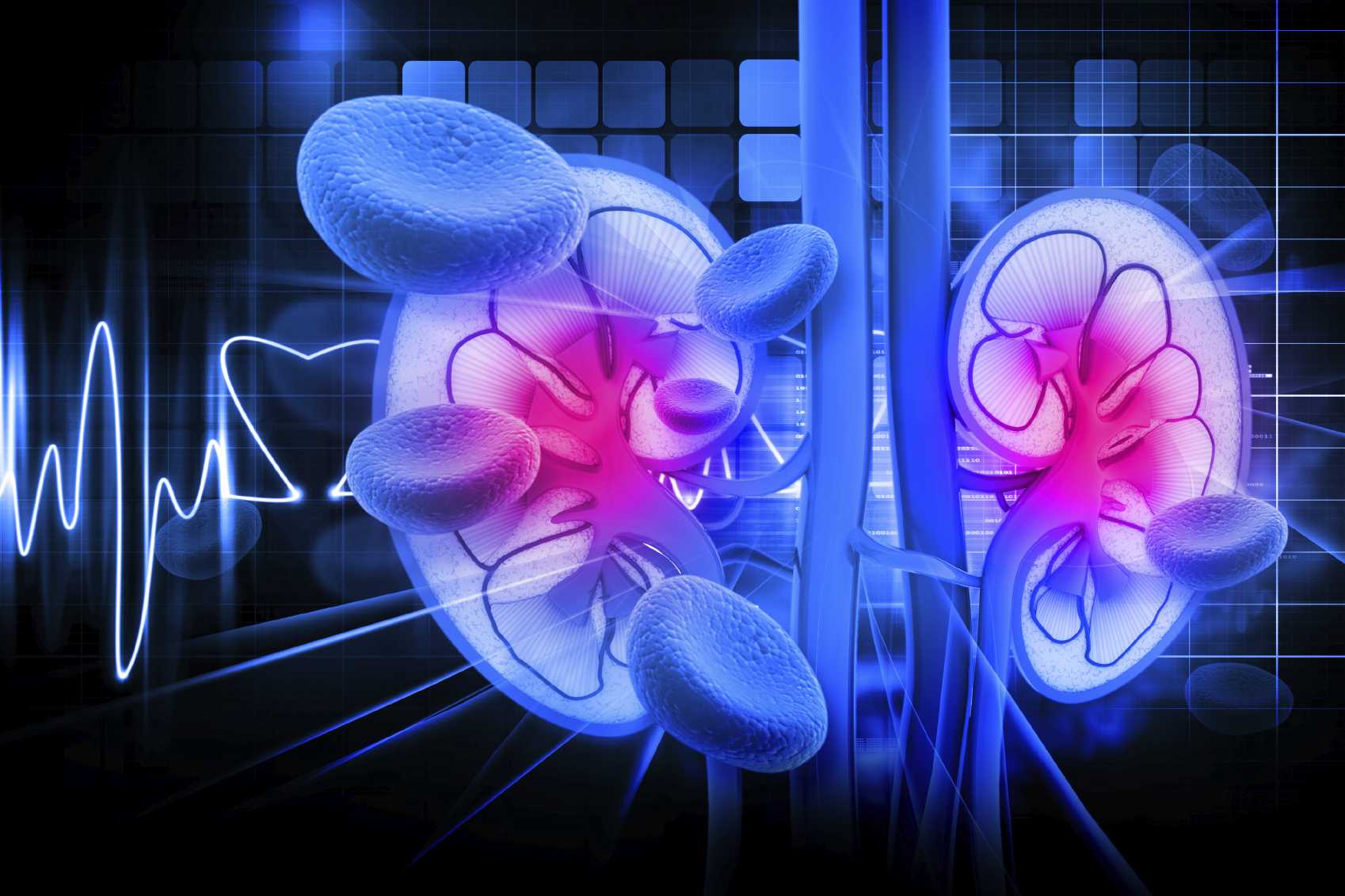
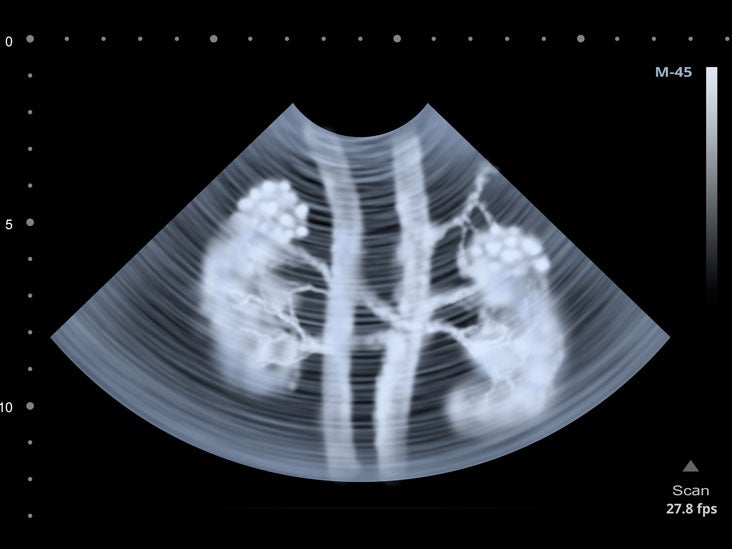


FIGURE 1.1: BASIC PICTURE OF KIDNEYS

2

**1.2 AIM OF THE PROJECT**

The main goal of this project is to prevent progression of CKD (Chronic Kidney Disease) and kidney failure. That is to determine whether or not a patient is at risk of developing a chronic disease. The best way to do this is to diagnose CKD early and control the underlying cause,and to do so, the symptoms identification of CKD will be reviewed by using classification techniques using such Convolution Neural Network (CNN) and Mobilenet.



3

FIGURE 1.2: PICTURE OF KIDNEY DIAGNOSIS

### **1.3 METHODOLOGY**

A database from a recognized hospital or a medical facility will be taken as our initial data setWe would ensure that the dataset to contain both chronic and healthy images of the kidney specimen .The database would be divided into training dataset and test dataset based on the scale of accuracy.The dataset after acquisition can contain many images that includes processed and not processed images.Due to raw images there may be unwanted image area which is not significant for the image under consideration.

The image can also contain unwanted noise of many types.

The images will be resized and cropped to remove less significant parts.Noise will be reduced using Gaussian filter and median filter.Other features of the image like brightness, size, contrast would be adjusted according to the type of image. After all the pre-processing techniques, the images would be randomly divided into training set(70% images) and test set(30% images).The CNN algorithm refines its understanding with each layer. Each layer of CNN algorithm detects different features of an image to produce an output that gets better with each layer.In order to check and identify features that specifically reflect the input item, the complexity of the filters increases with each additional layer.

The input image is processed through a number of different filters during convolution. Each filter performs its function by turning on specific aspects of the image, after which it sends its output to the filter in the subsequent layer. The operations are repeated for dozens, hundreds, or even thousands of layers as each layer learns to recognize various features.After the evaluation of all the training cases, test cases are run through the algorithm.The algorithm searches for the desirable features and classifies the image whether it is chronic part or healthy part of the kidney tissue.This process is continued with each image and the final results are generated.

The input image is processed through a number of different filters during convolution. Each filter performs its function by turning on specific aspects of the image, after which it sends its output to the filter in the subsequent layer.

4

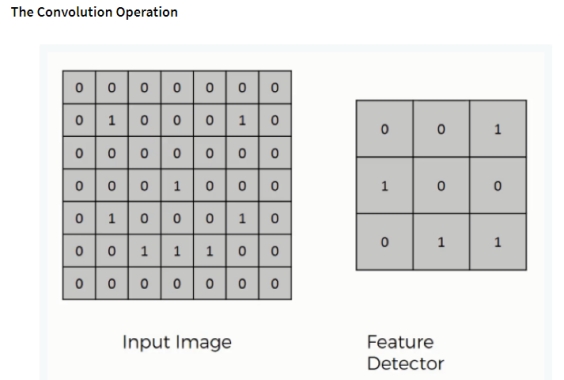
The operations are repeated for dozens, hundreds, or even thousands of layers as each layer learns to recognize various features.After the evaluation of all the training cases, test cases are run through the algorithm.The algorithm searches for the desirable features and classifies the image whether it is chronic part or healthy part of the kidney tissue.This process is continued with each image and the final results are generated.

**1.3.1 LAYERS USED IN PRE-TARINED MODEL – MOBILE NET**

**1. Convolutional Neural Network**

**Step(1a): convolutional operation**

The first building block in our plan of attack is convolution operation. In this step, we will touch on feature detectors, which basically serve as the neural network's filters. We will also discuss feature maps, learning the parameters of such maps, how patterns are detected, the layers of detection, and how the findings are mapped out.



5

FIGURE 1.3: DIAGRAM OF FEATURE DETECTOR FROM INPUT IMAGE

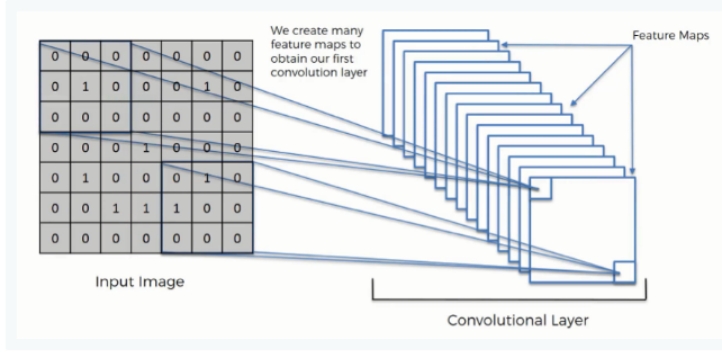


FIGURE 1.4: FORMATION OF CONVOLUTION LAYER FROM INPUT IMAGE

**Step (1b): ReLU Layer**

The second part of this step will involve the Rectified Linear Unit or Relook. We will cover Relook layers and explore how linearity functions in the context of Convolutional Neural Networks.

Not necessary for understanding CNN's, but there's no harm in a quick lesson to improve your skills.

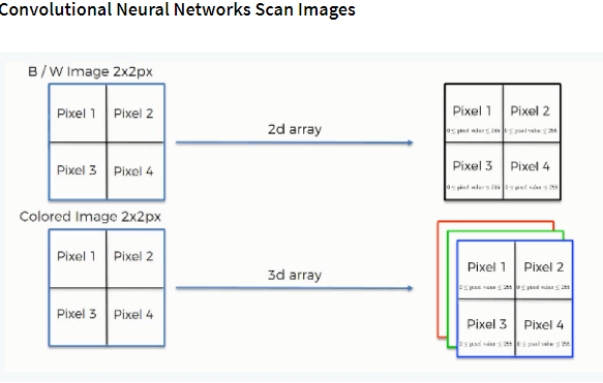


FIGURE 1.5: IMAGE OF RELU LAYER FORMATION

**Step 2: Pooling Layer**

In this part, we'll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We'll cover various approaches, though, including mean (or sum) pooling. This part will end with a demonstration made using a visual interactive

6

tool that will definitely sort the whole concept out for you.

The pooling operation involves sliding a two-dimensional filter over each channel of feature map and summarising the features lying within the region covered by the filter.   
For a feature map having dimensions **nh x nw x nc**, the dimensions of output obtained after a pooling layer is 

**(nh - f + 1)/ s x (nw - f + 1)/s x nc**

where,

* **nh -** height of feature map
* **nw -** width of feature map
* **nc -** number of channels in the feature map
* **f -** size of filter
* **s -** stride length

A common CNN model architecture is to have a number of convolution and pooling layers stacked one after the other.

#### ****Why to use Pooling Layers?****

* Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network.
* The pooling layer summarises the features present in a region of the feature map generated by a convolution layer. So, further operations are performed on summarised features instead of precisely positioned features generated by the convolution layer. This makes the model more robust to variations in the position of the features in the input image.

#### ****Types of Pooling Layers:****   ****Max Pooling****

1. Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

7

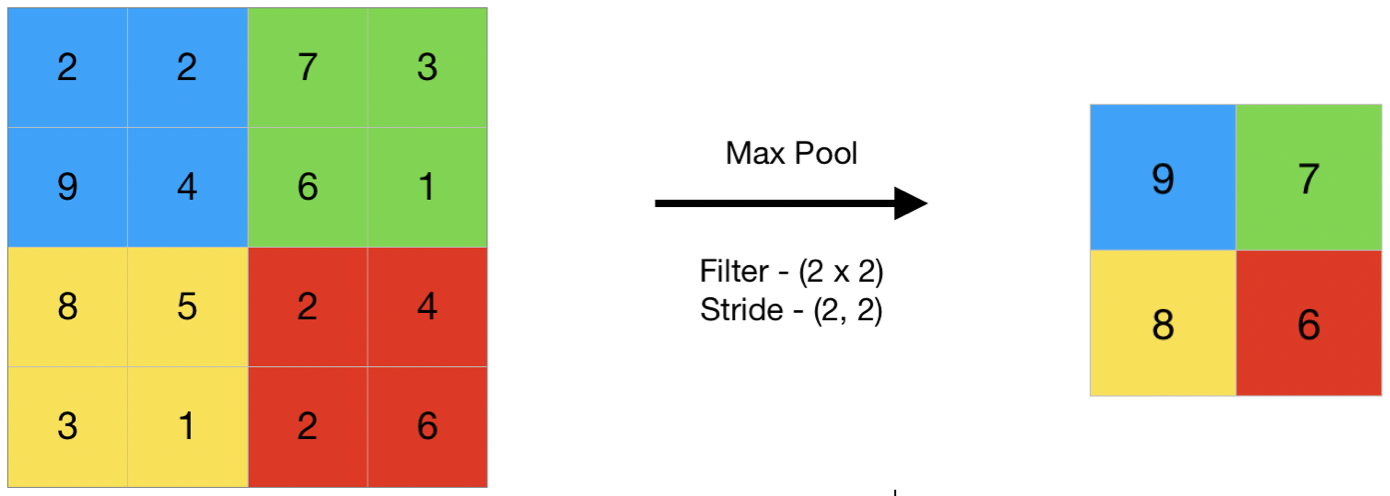


FIGURE 1.6:MAX POOLING OPERATION

### **Average Pooling**

1. Average pooling computes the average of the elements present in the region of feature map covered by the filter. Thus, while max pooling gives the most prominent feature in a particular patch of the feature map, average pooling gives the average of features present in a patch.

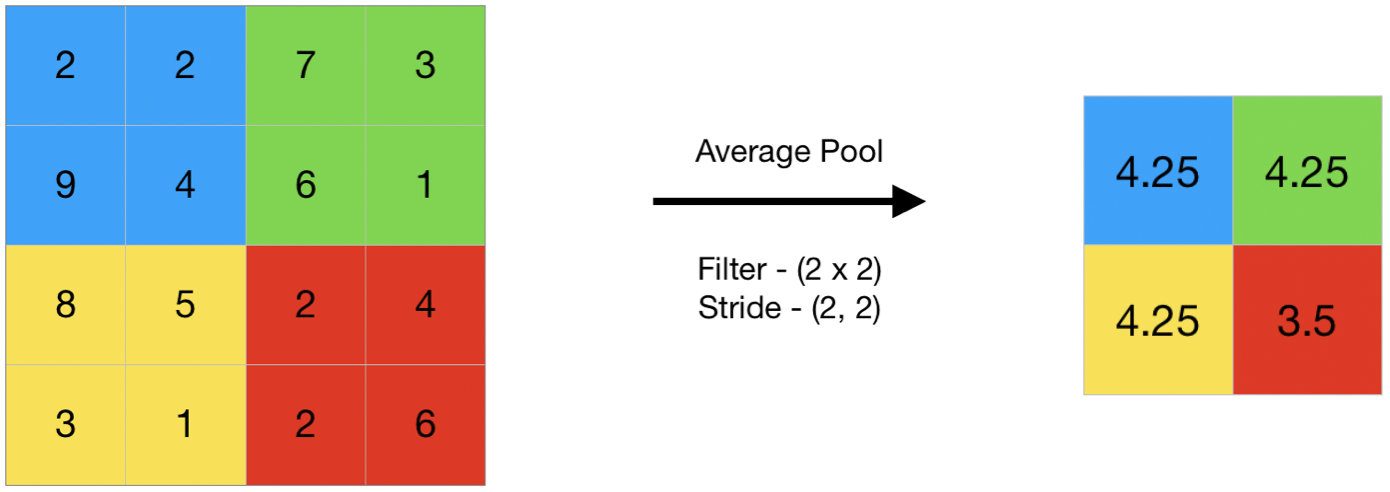


FIGURE 1.7: AVERAGE POOLING OPERATION

**Step 3: Flattening**

This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with Convolutional Neural Networks.

8

After finishing the previous two steps, we're supposed to have a pooled feature map by now. As the name of this step implies, we are literally going to flatten our pooled feature map into a column like in the image below.

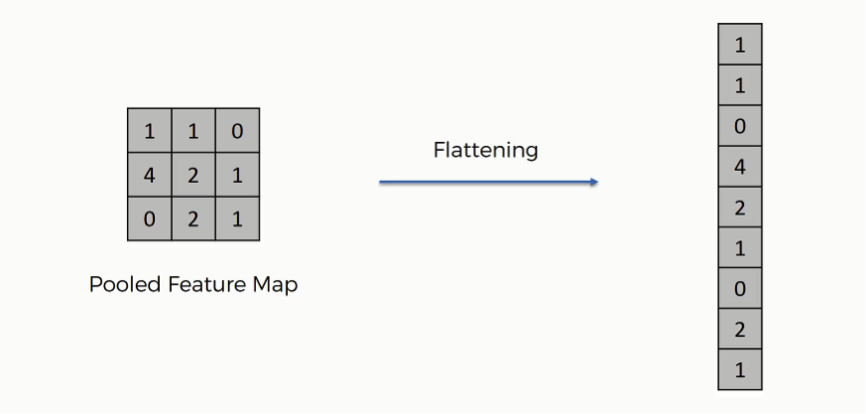


FIGURE 1.8: FLATTENING LAYER FROM FEATURE MAP

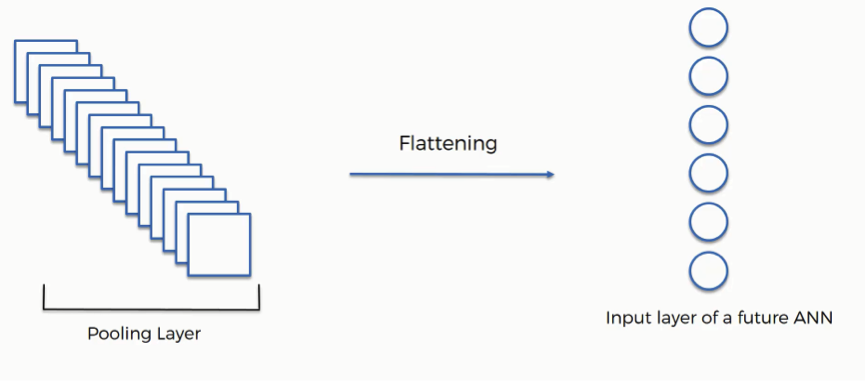


FIGURE 1.9: FLATTENED INPUT LAYER

As you see in the image above, we have multiple pooled feature maps from the previous step.

9

What happens after the flattening step is that you end up with a long vector of input data that you then

pass through the artificial neural network to have it processed further.

To sum up, here is what we have after we're done with each of the steps that we have covered up until now:

* Input image (starting point)
* Convolutional layer (convolution operation)
* Pooling layer (pooling)
* Input layer for the artificial neural network (flattening)

**Step 4: Full Connection**

In this part, everything that we covered throughout the section will be merged together. By learning this, you'll get to envision a fuller picture of how Convolutional Neural Networks operate and how the "neurons" that are finally produced learn the classification of images.

It's here that the process of creating a convolutional neural network begins to take a more complex and sophisticated turn.

As you see from the image below, we have three layers in the full connection step:

* Input layer
* Fully-connected layer
* Output layer

10

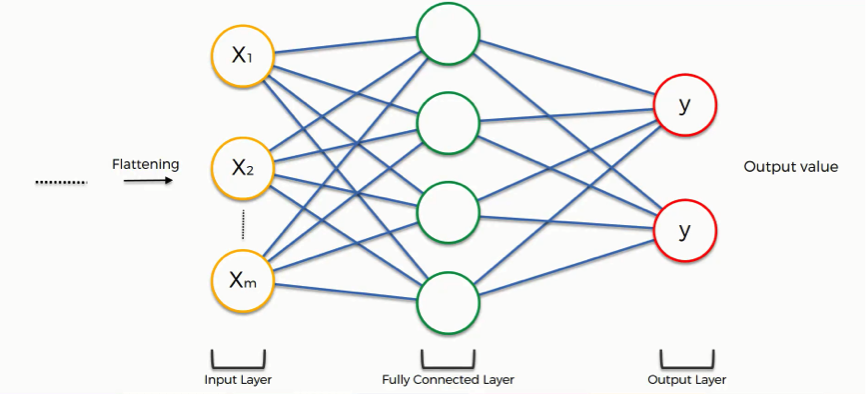


FIGURE 1.10: PICTURE OF FULLY CONNECTED LAYERS

Notice that when we discussed artificial neural networks, we called the layer in the middle a “hidden layer” whereas in the convolutional context we are using the term “fully-connected layer.”

**The Full Connection Process**

As we said in the previous tutorial, the input layer contains the vector of data that was created in the flattening step. The features that we distilled throughout the previous steps are encoded in this vector.

At this point, they are already sufficient for a fair degree of accuracy in recognizing classes. We now want to take it to the next level in terms of complexity and precision.

**What is the aim of this step?**

The role of the artificial neural network is to take this data and combine the features into a wider variety of attributes that make the convolutional network more capable of classifying images, which is the whole purpose from creating a convolutional neural network.

**Summary**

In the end, we'll wrap everything up and give a quick recap of the concept covered in the section. If you feel like it will do you any benefit (and it probably will), you should check out the extra tutorial in which

Softmax and Cross-Entropy are covered. It's not mandatory for the course, but you will likely come across these concepts when working with Convolutional Neural Networks and it will do you a lot of good to be familiar with them.

11

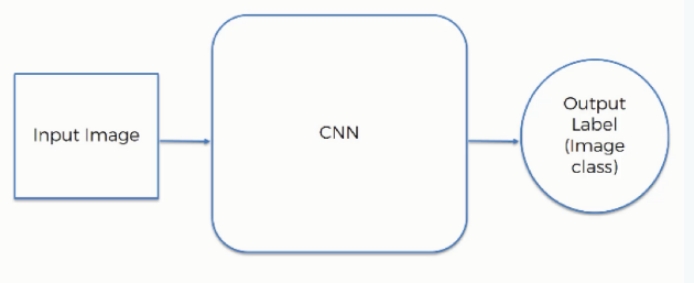


FIGURE 1.11: FULLY CONNECTED LAYER BLOCK DIAGRAM

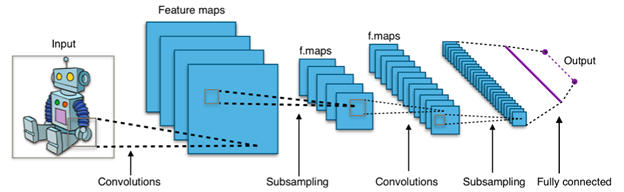


FIGURE 1.12: CONNECTION OF LAYERS IN CNN

**1.3.2 Pre-trained Model – MobileNet:**

MobileNet is an efficient and portable CNN architecture that is used in real world applications. MobileNets primarily use depth wise separable convolutions in place of the standard convolutions used in earlier architectures to build lighter models.MobileNets introduce two new global hyperparameters(width multiplier and resolution multiplier) that allow model developers to trade off latency or accuracy for speed and low size depending on their requirements.

12

What is MobileNet?

As the name applied, the MobileNet model is designed to be used in mobile applications, and it is TensorFlow’s first mobile computer vision model.

# **Image Classification with**MobileNet

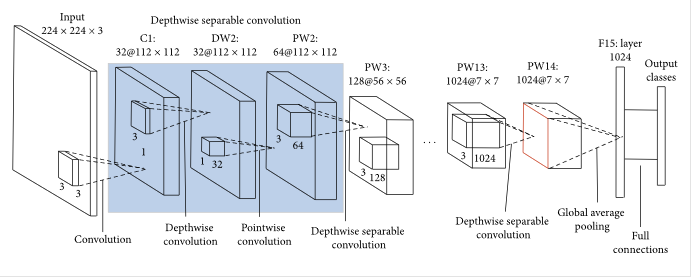


FIGURE 1.13: DIAGRAM OF PRE-TARINED MODEL

MobileNet uses depthwise separable convolutions. It significantly reduces the number of parameters when compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks.

A depthwise separable convolution is made from two operations.

1. Depthwise convolution.

2. Pointwise convolution.

MobileNet is a class of CNN that was open-sourced by Google, and therefore, this gives us an excellent starting point for training our classifiers that are insanely small and insanely fast.

Depthwise Separable Convolution

This convolution originated from the idea that a filter’s depth and spatial dimension can be separated- thus, the name separable. Let us take the example of Sobel filter, used in image processing to detect edges.

13

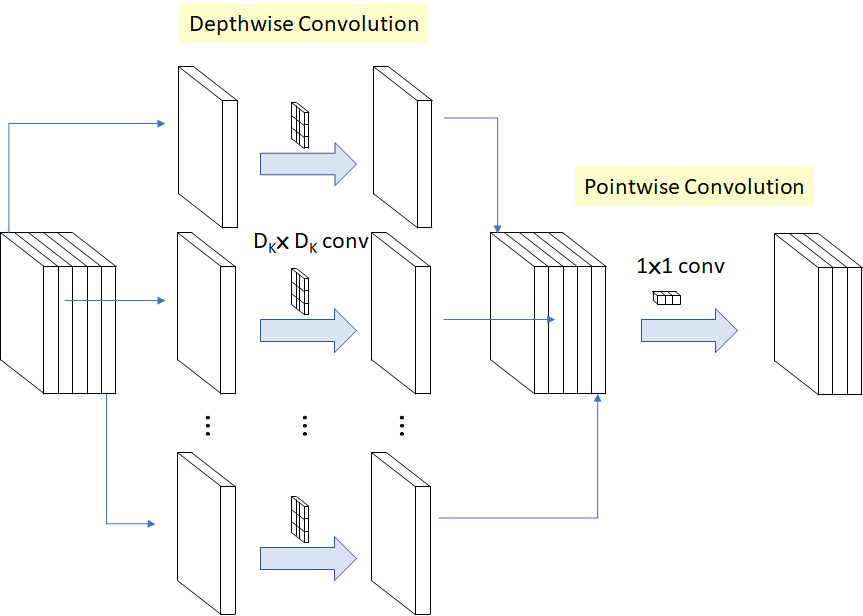


FIGURE 1.14: BLOCK DIAGRAM OF DEPTH WISE AND POINT WISE CONVOLUTION

You can separate the height and width dimensions of these filters. Gx filter can be viewed as a matrix product of [1 2 1] transpose with [-1 0 1].

We notice that the filter had disguised itself. It shows it had nine parameters, but it has 6. This has been possible because of the separation of its height and width dimensions.

The same idea applied to separate depth dimension from horizontal (width\*height) gives us depth-wise

separable convolution where we perform depth-wise convolution. After that, we use a 1\*1 filter to cover the depth dimension.

One thing to notice is how much parameters are reduced by this convolution to output the same no. of channels. To produce one channel, we need 3\*3\*3 parameters to perform depth-wise convolution and 1\*3

parameters to perform further convolution in-depth dimension.

14

But If we need three output channels, we only need 31\*3 depth filter, giving us a total of 36 parameters while for the same no. of output channels in regular convolution, we need 33\*3\*3 filters giving us a total of 81 parameters.

Depthwise separable convolution is a depthwise convolution followed by a pointwise convolution as follows:

Depthwise convolution is the channel-wise DK×DK spatial convolution. Suppose in the figure above, and we have five channels; then, we will have 5 DK×DK spatial convolutions.

Pointwise convolution is the 1×1 convolution to change the dimension.

1.Depthwise convolution.

It is a map of a single convolution on each input channel separately. Therefore its number of output channels is the same as the number of the input channels. Its computational cost is

Df² \* M \* Dk².

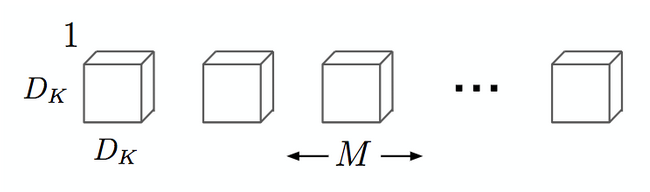


FIGURE 1.15: BLOCK DIAGRAM OF DEPTH WISE CONVOLUTION

2.pointwise convolution.

Convolution with a kernel size of 1x1 that simply combines the features created by the depthwise convolution. Its computational cost is

M \* N \* Df².

15

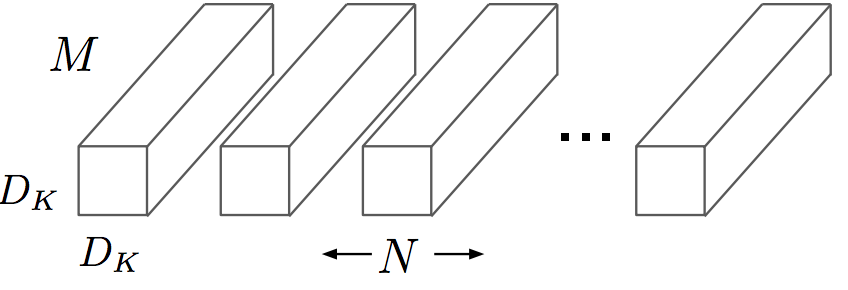


FIGURE 1.16: BLOCK DIAGRAM OF POINT WISE CONVOLUTION

MobileNets are a family of mobile-first computer vision models for TensorFlow, designed to effectively maximize accuracy while being mindful of the restricted resources for an on-device or embedded application.

MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use cases. They can be built upon for classification, detection, embeddings, and segmentation.

The main difference between MobileNet architecture and a traditional CNN instead of a single 3x3 convolution layer followed by the batch norm and ReLU. Mobile Nets split the convolution into a 3x3 depth-wise conv and a 1x1 pointwise conv, as shown in the figure.

**Architecture**

MobileNets are built on depth wise separable convolution layers. Each depth wise separable convolution layer consists of a depth wise convolution and a point wise convolution. Counting depth wise and point wise convolutions as separate layers, a MobileNet has 28 layers. A standard MobileNet has 4.2 million parameters which can be further reduced by tuning the width multiplier hyperparameter appropriately.

The size of the input image is 224 × 224 × 3.

A single standard convolution unit (denoted by Conv in the table above) looks like this:

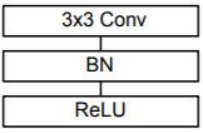


FIGURE 1.17: BLOCK DIAGRAM OF CONVOLUTION UNIT

UNIT BLOCK DIAGRAM

16

**SOFTWARE DEVELOPMENT LIFE CYCLE – SDLC:**

In our project we use waterfall model as our software development cycle because of its step-by-step procedure while implementing.

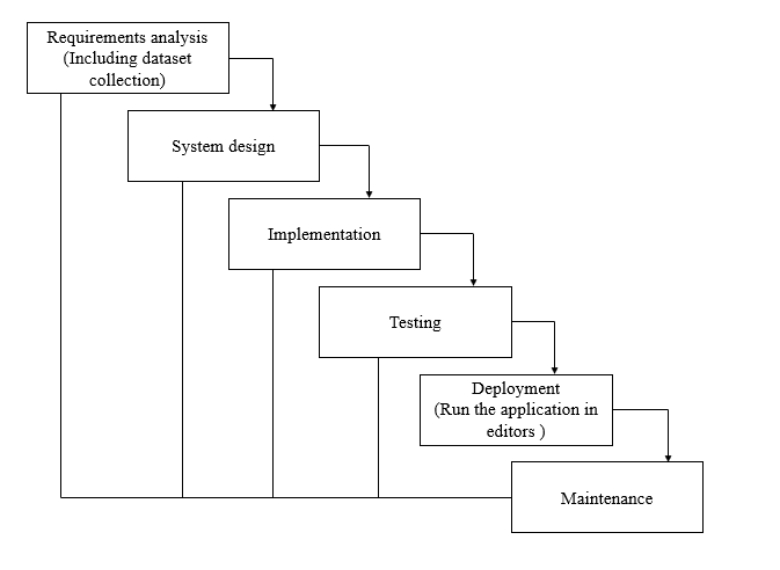


FIGURE 1.18: DIAGRAM OF SOFTWARE DEVLOPEMENT LIFE CYCLE

* **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − the requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system

17

requirements and helps in defining the overall system architecture.

* **Implementation** − with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

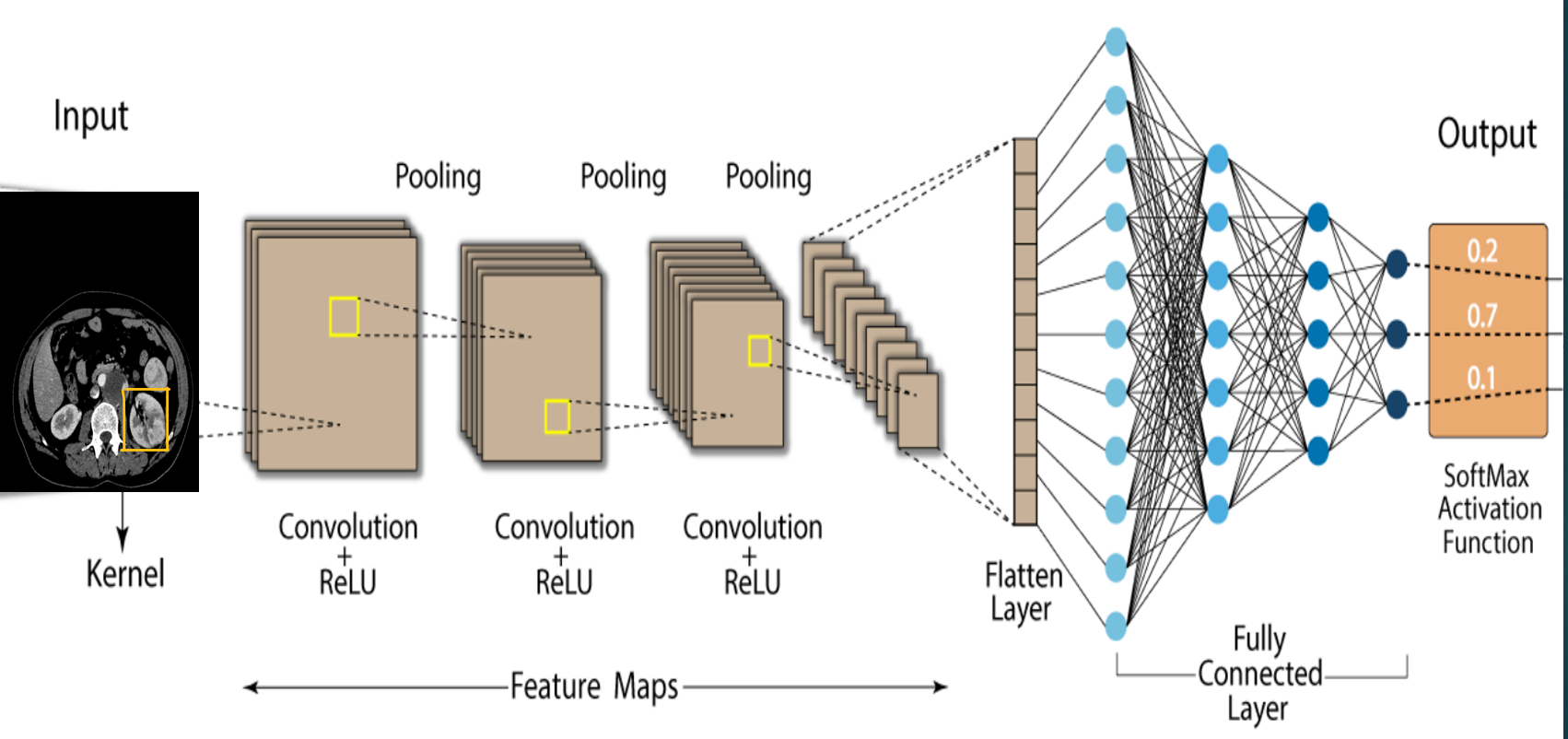
****

FIGURE 1.19: BLOCK DIAGRAM OF CNN ALGORITHM

18

**1.4 SIGNIFIACANCE AND APPLICATIONS**

**Significance:**

* Early detection of CKD enables patients to receive timely treatment to ameliorate the progression of this disease.
* High Accuracy.
* Performance is high.
* Time minimizing.
* Results are accurate.

**Applications:**

* This model can be used with not only the detection of chronic kidneys but also other body tissues like heart, brain, lungs etc., with proper data acquisition.
* This model can be combined with other health related models to create a multi purpose smart health monitoring system.
* This model can be embedded with highly precise robotic arms for highly accurate diagnosis without human intervention.

19

**1.5 ORGANISATION OF WORK**

CHAPTER 1

In this chapter we will briefly have an overview of the project consisting of:

Introduction of the smart irrigation system and need of the project, IOT technology, main Aim of the project, Methodology that is the technical approach for the project using sensors and Arduino uno consisting of block diagram, work flow explaining the process of implementation of the project, significance of the project including the advantages and real- life applications where this system is used.

CHAPTER 2

This chapter is about the literature survey of the project consisting of the problem statement, existing system and its disadvantage, proposed system having better result than the existing system.

CHAPTER 3

In this chapter we include the system requirement specifications which are hardware requirements for the construction of circuit and software requirements (software that we have used is Arduino IDE) and its installation steps.

CHAPTER 4

This chapter includes system analysis having the feasibility study of the project dividing into economical feasibility, operational feasibility, technical feasibiblity and system design having the circuit diagram and connections of the hardware components in the hardware design, programming language used, libraries used in the code.

CHAPTER 5

In this chapter we include final result having the smart irrigation system that is designed and the outputs from the sensors in the Arduino IDE, outputs from the bluetooth module to the Arduino Bluetooth terminal of the user’s device. Conclusion from the project and future scope of the project for the further development.

20

**CHAPTER 2. LITERATURE REVIEW**

**2.1 CASE STUDIES**

**[1] Z. Chen et al., “Diagnosis of patients with chronic kidney disease by using two fuzzy classifiers,” Chemometr. Intell. Lab., vol. 153, pp. 140-145, Apr.**

**2016.**

The feasibility of two in-house fuzzy classifiers, fuzzy rule-building expert system (FuRES) and fuzzy optimal associative memory (FOAM), for diagnosis of patients with chronic kidney disease (CKD) was investigated. A linear classifier, partial least squares discriminant analysis (PLS-DA), was used for comparison. The CKD data used in this work were taken from the UCI Machine Learning Repository. Composite datasets were created by adding different levels of proportional noise to evaluate the robustness of the two fuzzy approaches. Firstly, 11 levels of proportional noises were added to each numeric attribute of the training and prediction sets one after another, and then these simulated training and prediction sets were combined in pairs. Thus, a grid with 121 groups of simulated data was generated, and classification rates for these 121 pairs were compared. Secondly, the performances of two fuzzy classifiers using the simulated datasets, in which 11 levels of noise were randomly distributed to each numeric attribute, were compared and the average prediction rates of FuRES and FOAM were 98.1 ± 0.5% and 97.2 ± 1.2%, respectively, with 200 bootstrap Latin partitions. The PLS-DA can give 94.3 ± 0.8% with the identical evaluation. Confluent datasets comprised of the original and modified datasets were also used to evaluate FuRES, FOAM, and PLS-DA classification models. The average prediction rates of FuRES and FOAM obtained from 200 bootstrapped evaluations were 99.2 ± 0.3% and 99.0 ± 0.3%. PLS-DA yields slightly worse accuracy with 95.9 ± 0.6%. The results demonstrate that both FuRES and FOAM perform well on the identification of CKD patients, while FuRES is more robust than FOAM. These two fuzzy classifiers are useful tools for the diagnosis of CKD patients with satisfactory robustness, and can also be used for other kinds of patients

**Summary**: In the present work, two fuzzy classifiers, FuRES and FOAM, are applied for the classification of chronic kidney disease (CKD) patients. Based on the CKD data cited from the UCI Machine Learning website, their feasibility and robustness were investigated

**[2] C. Barbieri et al., “A new machine learning approach for predicting the response to anemia treatment in a large cohort of end stage renal disease patients undergoing dialysis,” Comput. Biol. Med., vol. 61, pp. 56-61, Jun. 2015.**

21

Chronic Kidney Disease (CKD) anemia is one of the main common comorbidities in patients undergoing End Stage Renal Disease (ESRD). Iron supplement and especially Erythropoiesis Stimulating Agents (ESA) have become the treatment of choice for that anemia. However, it is very complicated to find an adequate treatment for every patient in each particular situation since dosage guidelines are based on average behaviors, and thus, they do not take into account the particular response to those drugs by different patients, although that response may vary enormously from one patient to another and even for the same patient in different stages of the anemia. This work proposes an advance with respect to previous works that have faced this problem using different methodologies (Machine Learning (ML), among others), since the diversity of the CKD population has been explicitly taken into account in order to produce a general and reliable model for the prediction of ESA/Iron therapy response. Furthermore, the ML model makes use of both human physiology and drug pharmacology to produce a model that outperforms previous approaches, yielding Mean Absolute Errors (MAE) of the Hemoglobin (Hb) prediction around or lower than 0.6 g/dl in the three countries analyzed in the study, namely, Spain, Italy and Portugal.

**Summary:** This paper has presented a reliable ML approach to predict Hb values in patients undergoing secondary anemia to CKD. The work is the result of a long experience of the authors in this problem, with some previous works in which the produced models were not completely satisfactory.

**[3] V. Papademetriou et al., “Chronic kidney disease, basal insulin glargine, and health outcomes in people with dysglycemia: The origin study,” Am. J. Med., vol. 130, no. 12, Dec. 2017.**

Early stages of chronic kidney disease are associated with an increased cardiovascular risk in patients with established type 2 diabetes and macro vascular disease. The role of early stages of chronic kidney disease on macro vascular outcomes in prediabetes and early type two diabetes mellitus is not known. In the Outcome Reduction with an Initial Glargine Intervention (ORIGIN) trial, the introduction of insulin had no effect on cardiovascular outcomes compared to standard therapy. In this post hoc analysis of ORIGIN, we compared cardiovascular outcomes in subjects without to those with mild (Stages 1-2) and/or moderate chronic kidney disease (Stage 3). Methods. Two co-primary composite cardiovascular outcomes were assessed. The first was the composite endpoint of non-fatal MI, non-fatal stroke, or death from cardiovascular causes; and the second was a composite of any of these events plus a revascularization procedure, or hospitalization for heart failure. Several secondary outcomes were pre-specified including micro vascular outcomes, incident diabetes, hypoglycaemia, weight, and cancers. Complete renal function data were available in 12,174 out of 12,537 ORIGIN participants.

22

A total of 8,114 had no chronic kidney disease (67%) while 4,060 had chronic kidney disease stage 1-3 (33%). When compared to non-CKD participants, the risk of developing the composite primary outcome (nonfatal myocardial infarction, nonfatal stroke, or cardiovascular death) in those with mild to moderate chronic kidney disease was 87% higher; hazard ratio (HR): 1.87; 95% confidence interval (CI): 1.71-2.04 (p<0.001).

**Summary:** In high-risk patients with dysglycemia (pre-diabetes and early diabetes), mild and moderate chronic kidney disease significantly increased cardiovascular events.

### **2.2 EXISTING SYSTEM**

In the existing system, implementation of machine learning algorithms is bit complex to build due to the lack of information about the data visualization. Here the process is performed using the logistic regression, random forest, support vector machine, k-nearest neighbor, Naive Bayes classifier and feed forward neural network where these algorithms are unable to perform accurately and couldn’t get the proper accuracy. Mathematical calculations are used in existing system for model building this may takes the lot of time and complexity. To overcome all this, we use machine learning packages available in the scikit-learn library**.**

**Disadvantages:**

* High complexity.
* Time consuming.
* It is difficult to detect and identify diseases.
* Difficult to handle.
* Difficult to scale up.
* Low accuracy

### **2.3 PROPOSED SYSTEM**

Proposed several machine learning models in this section, the classifiers were first established by different deep learning algorithms to diagnose the data samples. Among these models, those with better

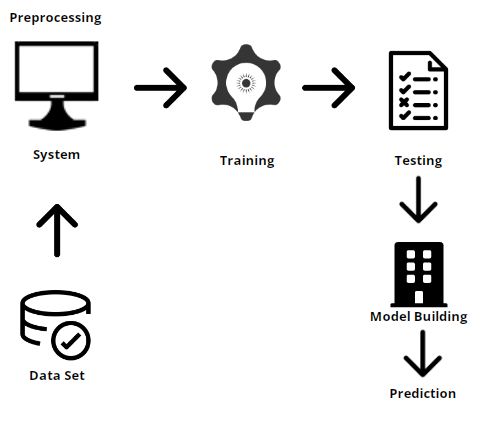
performance were selected as potential components. By analyzing their misjudgments, the component

models were determined. An integrated model was then established to achieve higher performance.

23

**Advantages:**

* High Accuracy.
* Performance is high.
* Time minimizing.
* Results are accurate.
* Reduces time complexity.
* Easy to handle.
* As we trained our model using the CNN algorithm we get the accurate results during the disease prediction.
* This work does not requires much efforts to be performed and can performed easily. So, it is easy to handle.



24

FIGURE 2.1: STEPS INVOLVED IN PROPOSED SYSTEM

# 

FIGURE 2.2: WORK FLOW OF THE PROJECT

25

**WORK FLOW**

**Create Dataset:**

The collection comprising data of the categorized disease type of CT Kidney disorders.

**Image Pre-processing:**

The images are resized and reshaped into the proper format to train our model.

**Splitting Data:**

Data splitting into training and test datasets with a test size of 30%–20%.

**Training:**

The CNN Deep learning algorithm and Mobile Net transfer learning techniques are utilized to train our model utilizing the pre-processed training dataset.

**Classification:**

The output of our algorithm is a display of CT pictures of kidney disorders that either have different labels or none at all.

**Upload Image:**

The user must submit an image for classification.

**View Results:**

The viewer views the results for the categorized images.

26

# **CHAPTER 3. SYSTEM REQUIREMENT SPECIFICATIONS**

**3.1 SOFTWARE AND HARDWARE REQUIREMENTS:**

**Hardware:**

Operating system : Windows 7 or 7+

RAM : 8 GB

Hard disc or SSD : More than 500 GB

Processor : Intel 3rd generation or high or Ryzen with 8 GB Ram

**Software:**

Software’s : Python 3.6 or high version

IDE : PyCharm, VISUAL STUDIO CODE

Framework : Flask

Technology/Language : Python

**Functional and non-functional requirements:**

Requirement’s analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

**Functional Requirements**: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1. Authentication of user whenever he/she logs into the system
2. System shutdown in case of a cyber-attack
3. A verification email is sent to user whenever he/she register for the first time on some software system.

**Non-functional requirements**: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

27

They basically deal with issues like:

* Portability
* Security
* Maintainability
* Reliability
* Scalability
* Performance
* Reusability
* Flexibility

Examples of non-functional requirements:

1. Emails should be sent with a latency of no greater than 12 hours from such an activity.
2. The processing of each request should be done within 10 seconds
3. The site should load in 3 seconds whenever of simultaneous users are > 10000

1.OPTIMIZER USED - Adam optimizer

2.LOSS FUNCTION USED - Root mean square propagation

3.NUMBER OF FILRTERS USED (IN CONVOLUTION LAYER) - 32 convolution filters

4.TOTAL NUMBER OF FILTERS USED - 32 convolution+ 64 max poolling + 96 RELU layers

5.96 RELU ACTIVATION FUNCTIONS

6.SOFTWARE USED: VISUAL STUDIO CODE

**Adam Optimizer**

Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the ‘gradient descent with momentum’ algorithm and the ‘RMSP’ algorithm.

**How Adam works?**

Adam optimizer involves a combination of two gradient descent methodologies:

**Momentum:**

This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the ‘exponentially weighted average’ of the gradients. Using averages makes the algorithm converge towards the minima in a faster pace.



where,



28

mt = aggregate of gradients at time t [current] (initially, mt = 0)

mt-1 = aggregate of gradients at time t-1 [previous]

Wt = weights at time t

Wt+1 = weights at time t+1

αt = learning rate at time t

∂L = derivative of Loss Function

∂Wt = derivative of weights at time t

β = Moving average parameter (const, 0.9)

**Root Mean Square Propagation (RMSP):**

Root mean square prop or RMSprop is an adaptive learning algorithm that tries to improve AdaGrad. Instead of taking the cumulative sum of squared gradients like in AdaGrad, it takes the ‘exponential moving average’.



where,



Wt = weights at time t

Wt+1 = weights at time t+1

αt = learning rate at time t

∂L = derivative of Loss Function

∂Wt = derivative of weights at time t

Vt = sum of square of past gradients. [i.e sum(∂L/∂Wt-1)] (initially, Vt = 0)

β = Moving average parameter (const, 0.9)

ϵ = A small positive constant (10-8)

**Mathematical Aspect of Adam Optimizer**



**Performance:**

Building upon the strengths of previous models, Adam optimizer gives much higher performance than the previously used and outperforms them by a big margin into giving an optimized gradient descent. The plot is shown below clearly depicts how Adam Optimizer outperforms the rest of the optimizer by a considerable margin in terms of training cost (low) and performance (high).

29

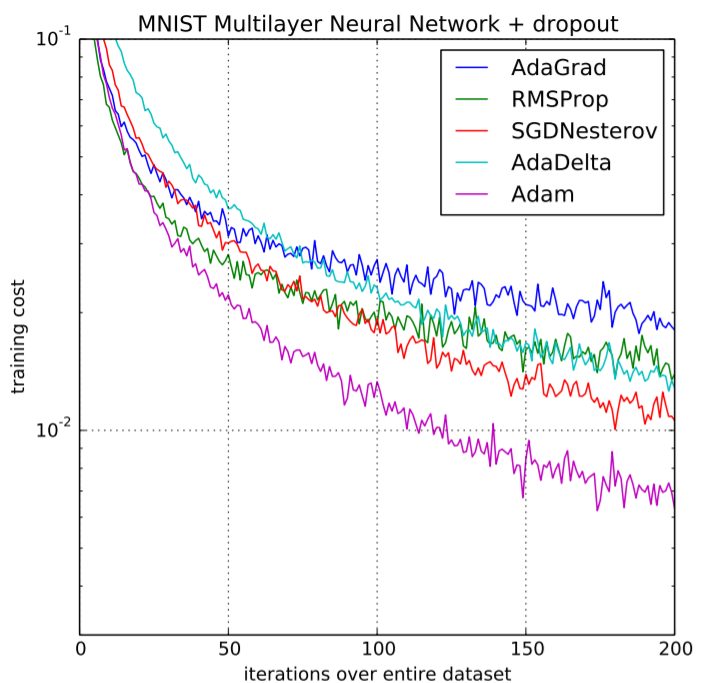


FIGURE 3.1: PERFORMANCE COMPARISION ON TRAINING COST

30

# **CHAPTER 4. SYSTEM ANALYSIS AND SYSTEM DESIGN**

**4.1 FEASIBILITY STUDY**

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**Economic feasibility:**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **Technical feasibility:**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**Social feasibility:**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

31

**4.2 SYSTEM DESIGN:**

## Input Design:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding −

### Objectives for Input Design:

The objectives of input design are −

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.
* To use validation checks and develop effective input controls.

**Output Design:**

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

### Objectives of Output Design:

The objectives of input design are:

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end user’s requirements.

32

* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions.

**DFD DIAGRAM:**

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and

boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

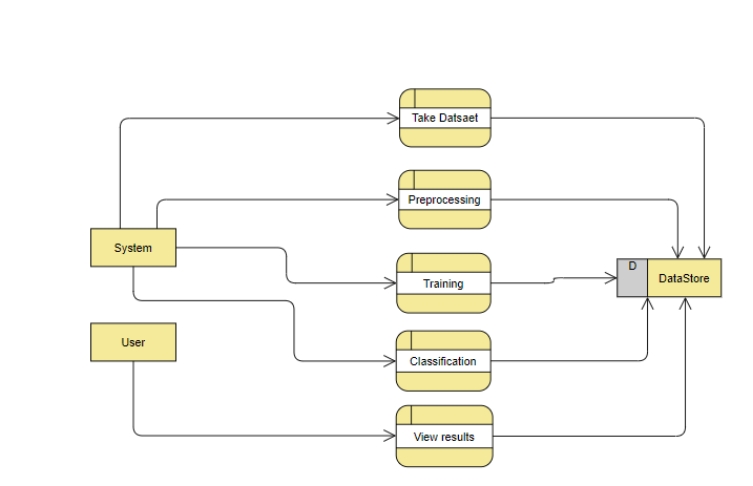


FIGURE 4.1: DIAGRAM OF DFD STEP 1

33



FIGURE 4.2: DIAGRAM OF DFD STEP 2

**TEST CASES:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Result** |
| Input text | Tested for the classification of person affected with kidney disease or not | Success |

TABLE 4.1: TABLE OF TEST CASES

34

**Test cases Model building:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Test cases** | **I/O** | **Expected O/T** | **Actual O/T** | **P/F** |
| 1 | Read the dataset. | Dataset path. | Dataset need to read successfully. | Dataset fetched successfully. | P |
| 2 | Performing pre-processing on the dataset | Pre-processing part takes place | Pre-processing should be performed on dataset | Pre-processing successfully completed. | P |
| 3 | Model Building | Model Building for the clean data | Need to create model using required algorithms | Model Created Successfully. | P |
| 4 | Classification | Input image provided. | Output should be the either pneumonia affected or normal classification | Model classified successfully | P |

TABLE 4.2: TABLE OF TEST CASES MODEL BUILDING

35

**CHAPTER 5. RESULTS AND CONCLUSIONS**

**5.1 RESULTS**

**5.1.1 Results from CNN:**

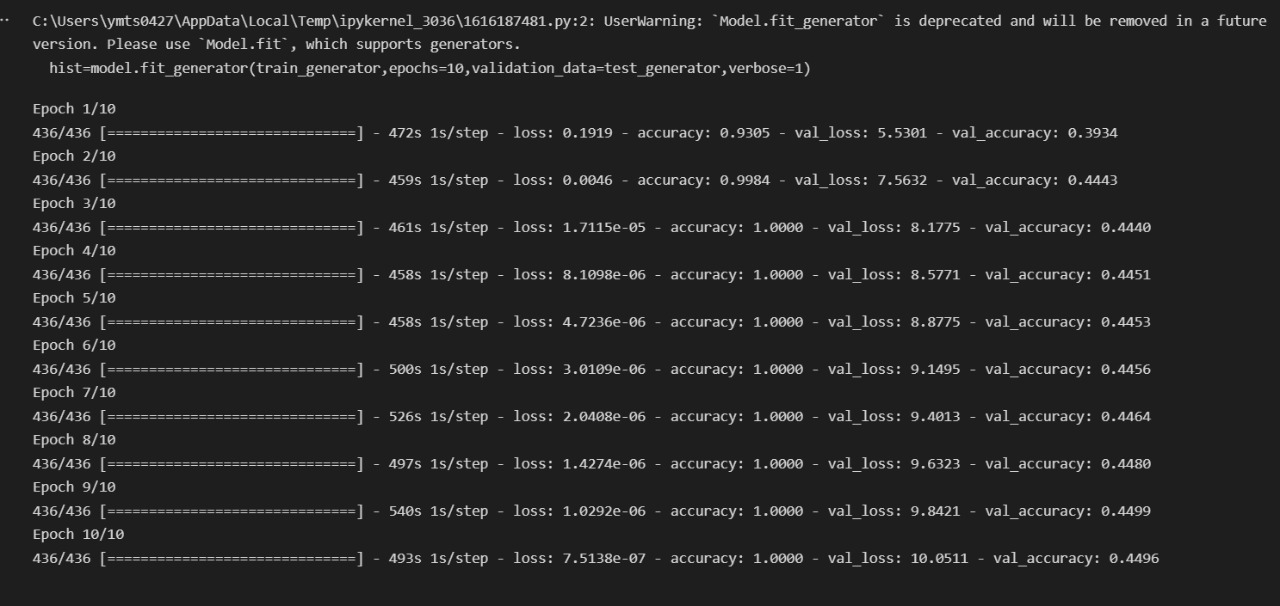
****

FIGURE 5.1: RESULT FROM CNN

36

****

FIGURE 5.2: GRAPH OF RESULT FROM CNN

37

**5.1.2 Results from Pre-trained model:**

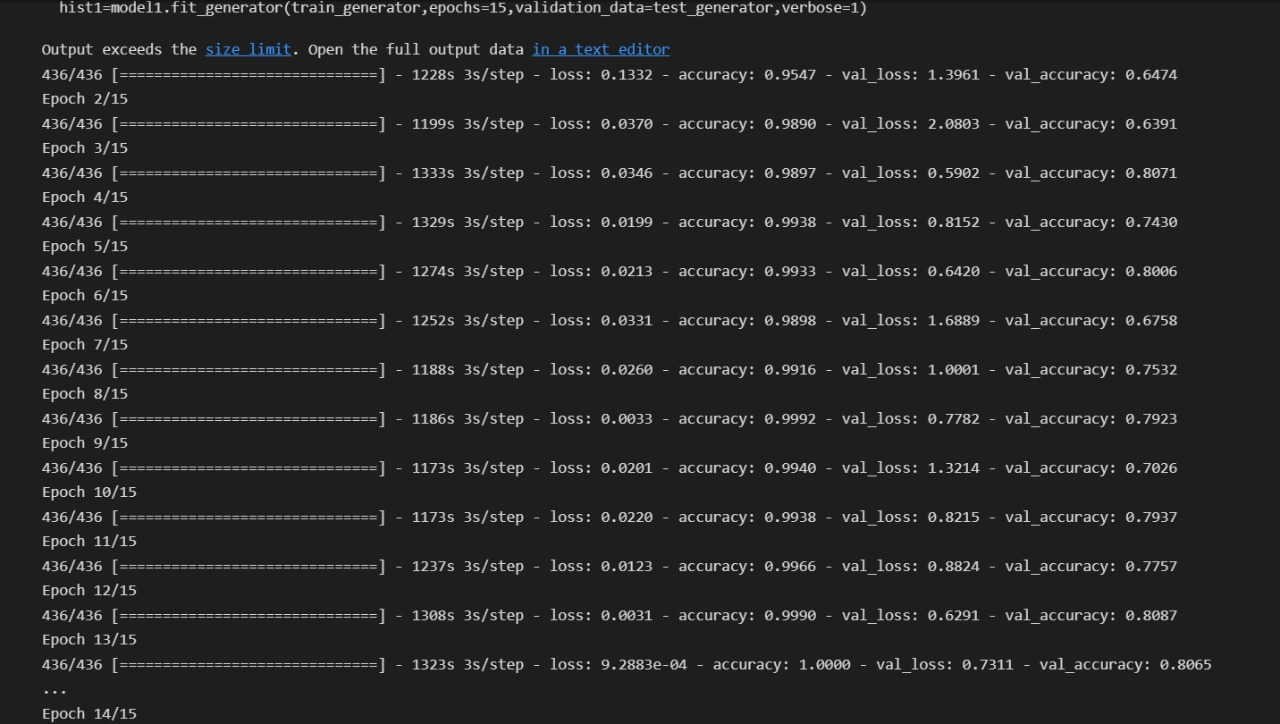
****

FIGURE 5.3: RESULT FROM PRE-TRAINED MODEL

38

#### 

FIGURE 5.4: GRAPH OF RESULT FROM PRE-TRAINED MODEL

39

#### 

#### 5.2: CONCLUSION AND FUTURE SCOPE

**Conclusion**:

The proposed CKD diagnostic methodology is feasible in terms of data imputation and samples diagnosis. After unsupervised imputation of missing values in the data set by using CNN imputation, the integrated model could achieve a satisfactory accuracy. Hence, we speculate that applying this methodology to the practical diagnosis of CKD would achieve a desirable effect. In addition, this methodology might be applicable to the clinical data of the other diseases in actual medical diagnosis. However, in the process of establishing the model, due to the limitations of the conditions, the available data samples are relatively small, therefore, the generalization performance of the model might be limited.

The main objective of this study was to predict patients with CKD using less number attributes while maintaining a higher accuracy.

Here we obtain an accuracy of about 96 percentage.

**Future scope:**

In the future, a large number of more complex and representative data will be collected to train the model to improve the generalization performance while enabling it to detect the severity of the disease. We believe that this model will be more and more perfect by the increase of size and quality of the data.

40

## REFERENCES

[1] Z. Chen et al., “Diagnosis of patients with chronic kidney disease by using two fuzzy classifiers,” Chemometr. Intell. Lab., vol. 153, pp. 140-145, Apr. 2016.

[2] L. Zhang et al., “Prevalence of chronic kidney disease in china: a cross-sectional survey,” Lancet, vol. 379, pp. 815-822, Aug. 2012.

[3] A. Singh et al., “Incorporating temporal EHR data in predictive models for risk stratification of renal function deterioration,” J. Biomed. Inform. vol. 53, pp. 220-228, Feb. 2015.

[4] H. Polat, H.D. Mehr, A. Cetin, “Diagnosis of chronic kidney disease based on support vector machine by feature selection methods,” J. Med. Syst., vol. 41, no. 4, Apr. 2017.

[5] C. Barbieri et al., “A new machine learning approach for predicting the response to anemia treatment in a large cohort of end stage renal disease patients undergoing dialysis,” Comput. Biol. Med., vol. 61, pp. 56-61, Jun. 2015.

[6] V. Papademetriou et al., “Chronic kidney disease, basal insulin glargine, and health outcomes in people with dysglycemia: The origin study,” Am. J. Med., vol. 130, no. 12, Dec. 2017.

[7] H. S. Xu, L.Wang, andW. L. Gan, Application of improved decision tree method based on rough set in building smart medical analysis CRM system, Int. J. Smart Home, vol. 10, no. 1, pp. 251–266, 2016.

[8] N. R. Hill et al., “Global prevalence of chronic kidney disease – A systematic review and meta-analysis,” Plos One, vol. 11, no. 7, Jul. 2016.

[9] M. M. Hossain et al., “Mechanical anisotropy assessment in kidney cortex using ARFI peak displacement: Preclinical validation and pilot in vivo clinical results in kidney allografts,” IEEE Trans. Ultrason. Ferr. vol. 66, no. 3, pp. 551-562, Mar. 2019.

[10] M. Alloghani et al., “Applications of machine learning techniques for software engineering learning and early prediction of students’ performance,” in Proc. Int. Conf. Soft Computing in Data Science, Dec. 2018, pp. 246- 258.

41

**Web links:**

[11] website name – pubmed

<https://pubmed.ncbi.nlm.nih.gov/33328123/>

[12] website name – pubmed

<https://pubmed.ncbi.nlm.nih.gov/31573641/>

[13] website name – NIH

<https://www.niddk.nih.gov/health-information/kidney-disease/chronic-kidney-disease-ckd>

**Journals:**

[14] website name – IEEE Xplore

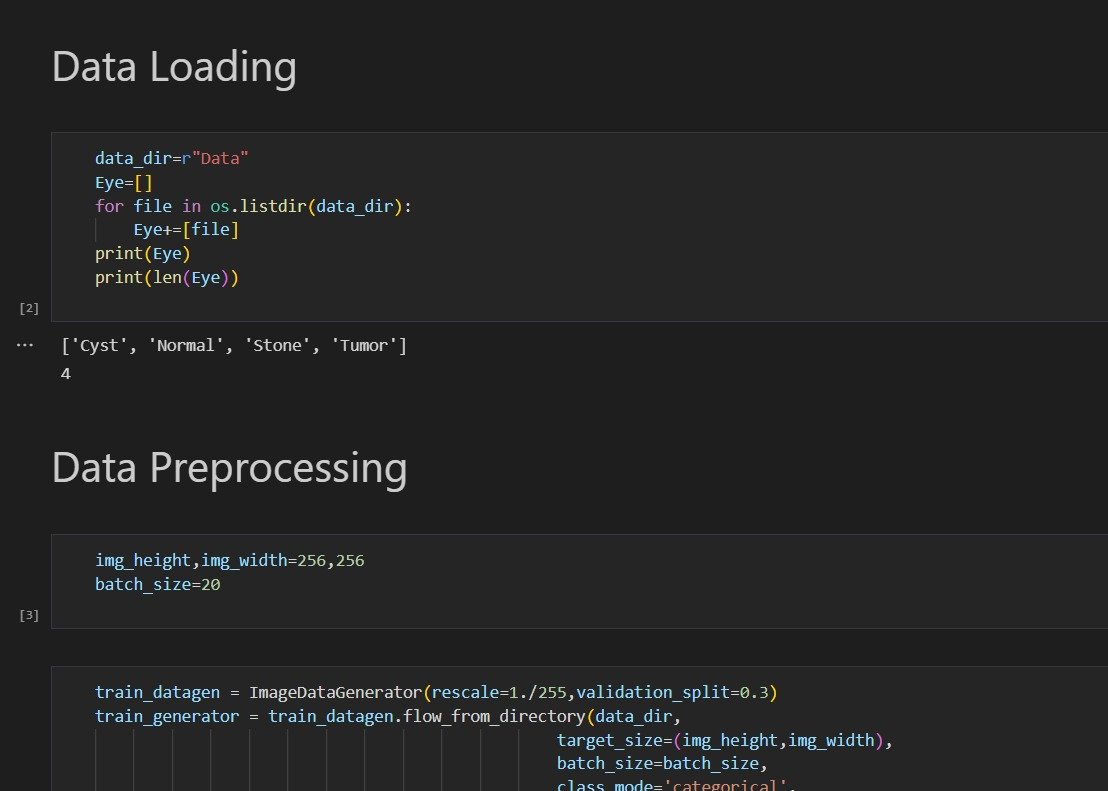
<https://ieeexplore.ieee.org/document/9623101>

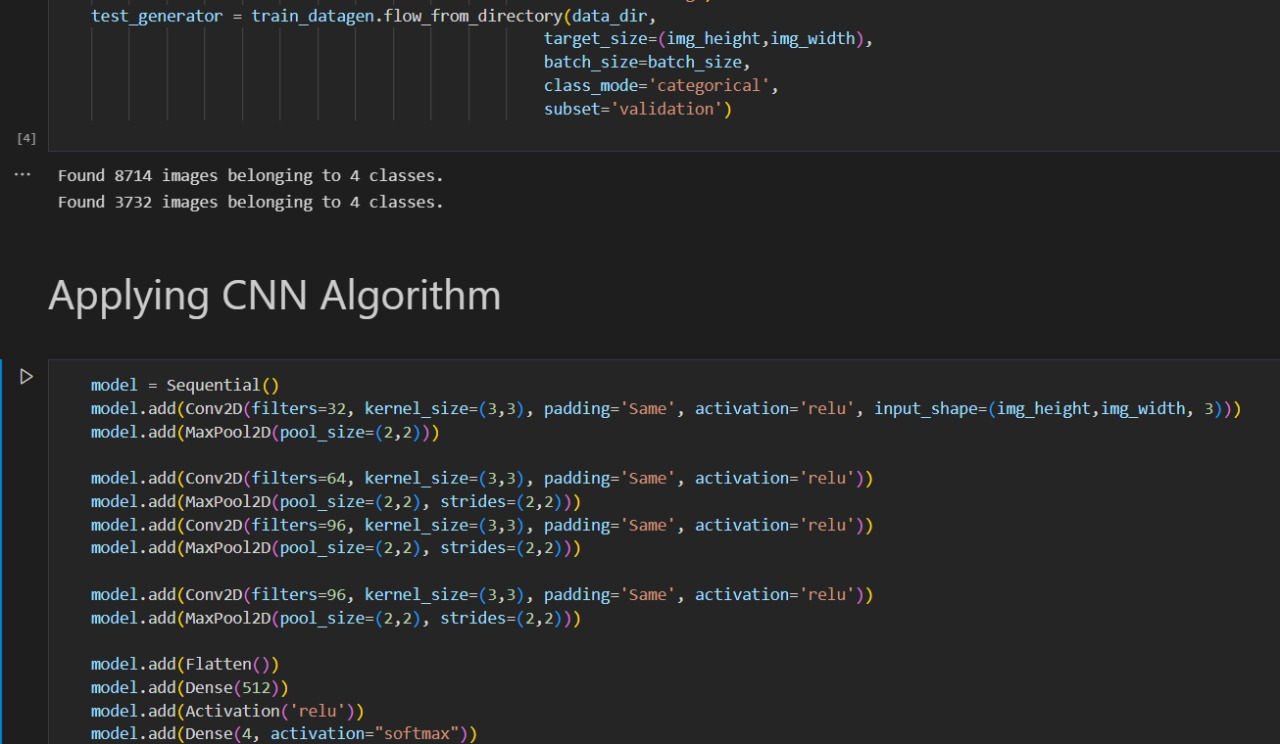
[15] website name – IJET

<https://www.ijert.org/chronic-kidney-disease-prediction-using-machine-learning>

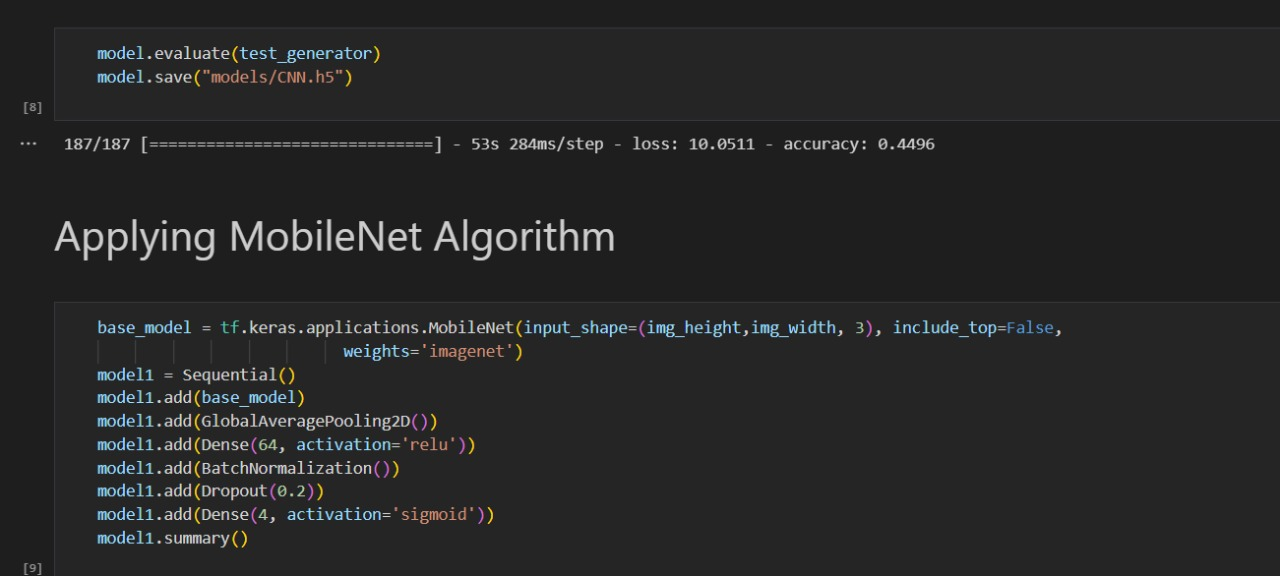
42

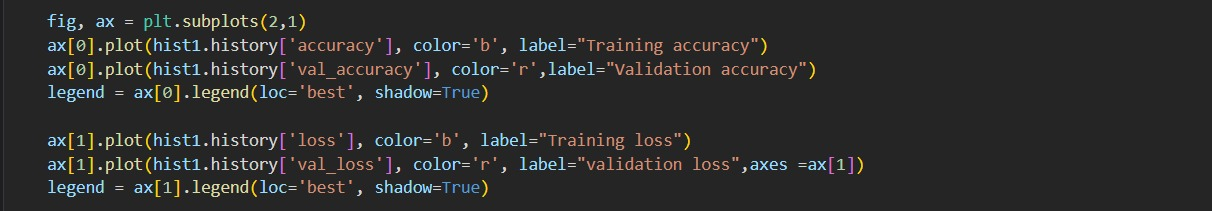
## APPENDIX





43





44





45